Assignment 4

Due: Sunday, October 28, 2018 (11:59PM)

Submission Instruction

For this assignment, you need to submit:

a) CSPSudokuSolver.java
b) solution_4x4.txt
c) solution_9x9.txt
d) experiment_result.txt

Upload your both files to the mycourse website. DO NOT zip your files.

Please note that there is a penalty for late submission: 10% of your score per hour (50% max).

1. CSPSudokuSolver.java is the only file that you will modify. At the main method, change studentID to your ID. Notice that there are 5 places marked with
   `// [start:x]`

   `// [end:x]`
   There are also a few options that you have to play with:
   ```
   public static int LIMIT = 9*9+1;  // helpful for debugging
   public static boolean MAC = false;
   public static boolean MRV = false;
   public static boolean INFER = false;
   public static boolean LCV = false;  // not required
   ```

2. Implement a basic backtrack search algorithm `[start:0]`. Test your code with the small `testBoard` (4x4) with all options set to false.

3. Implement variable ordering and filtering methods (block: 1 - 4). Test your code with the small `testBoard` (4x4).

4. Collect result. If everything works fine, put the sudoku solutions in their corresponding files (solution_4x4.txt and solution_9x9.txt). At the end of each file, report the runtimes for
    a. forward checking
    b. MAC
    c. forward checking + MRV
    d. MAC + MRV

    For each setting, you should run the solver five times and report the average runtime. You file will look like this:

```
 1 2 | 3 4
 3 4 | 1 2
-----------
 2 3 | 4 1
 4 1 | 2 3
8.5
5.5
6.5
7.3
```

5. Run experiment. Finally, run an `experiment(studentId);` and collect runtime for the same settings as (4).

   Your experiment_result.txt file will look like this:

```
2062216300
-1211614091
-1034775538
...
-304242310
-434417225
-2009276312
max,min,avg
5431,8,423.13
3334,3,223.97
3030,4,323.23
2030,2,123.43
```

   The first 200 lines will be a hash code of randomly selected boards and the last 4 lines are your experiment result.

   UPDATE: 14/10/2018: Solving 200 sudoku boards will take a very long time. You can try to solve 50 for the experiment instead. Just change `for (int i = 0; i < 200; i++){` to `for (int i = 0; i < 50; i++){`